

NOTE SULLO SVOLGIMENTO DELLA PROVA:

- Fare partire il Dev-C++ (dal Menù **Avvio** (o **Start**) nella barra degli strumenti in fondo allo schermo, selezionare Programmi e quindi Dev-C++);
- attraverso il menu **File->Open...** (o **apri** se il Dev-C++ è in italiano) aprire il progetto *APRIQUESTO.dev* presente nel directory *c:\esame\esaInf*;
- scrivere le funzioni richieste nel file *compito.cpp* già presente nel progetto;
- salvare spesso in modo da non perdere il lavoro nel caso in cui il PC abbia un malfunzionamento (per salvare premere il bottone a forma di dischetto del Dev-C++);
- per una corretta stampa dell'elaborato bisogna mantenere il codice entro i margini imposti dall'ambiente Dev-C++ (linea verticale presente alla destra della pagina).

Richieste asilo

La gestione di una coda di richieste per accedere ai posti di un asilo può essere realizzata a partire dalle seguenti costanti e dai seguenti tipi:

```
const int MAXS = 100;
const int MAXR = 200;

enum Genere {maschile, femminile};

struct Richiesta {
    char codfis[MAXS];
    float reddito;
    int numComp;
    int eta;
};

struct CodaRichieste {
    Richiesta ric[MAXR];
    int front, back;
};
```

Il tipo *Richiesta* memorizza i dati di una richiesta per un posto asilo: *codfis* è il codice fiscale (semplificato) del bambino, *reddito* indica il reddito complessivo del nucleo familiare, *numComp* rappresenta il numero di membri del nucleo familiare, *eta* indica il numero di anni del bambino per cui viene fatta la richiesta (i valori possibili vanno da 1 a 5 estremi inclusi). La struttura *CodaRichieste* mantiene le richieste secondo un ordinamento FIFO. Il tipo *Genere* viene usato per codificare il sesso del bambino per cui viene fatta richiesta.

Realizzare le seguenti funzioni:

bool inserisci(CodaRichieste* pc, Richiesta r) inserisce in fondo alla coda una nuova richiesta (*r*). Affinché la richiesta possa essere inserita non deve essere già presente nella coda una richiesta avente lo stesso codice fiscale. La funzione restituisce *true* se l'inserimento va a buon fine, *false* altrimenti.

void generaCodice(const char nome[], const char cognome[], int anno, Genere g, char dst[]) riempie l'array *dst* con il codice fiscale (stringa). Il codice fiscale è così composto:

- le prime tre lettere sono ricavate dalle consonanti del cognome; se il cognome contiene un numero di consonanti inferiore a tre, i caratteri mancanti sono posti uguali ad 'x';
- le successive tre lettere sono ricavate dalle consonanti del nome; se il nome contiene un numero di consonanti inferiore a tre, i caratteri mancanti sono posti uguali ad 'x';

- seguono le due cifre meno significative dell'anno di nascita;
- l'ultimo carattere è una 'm' se il genere è maschile, 'f' se femminile.

Per esempio il codice fiscale di mario rossi, nato nel 2008, di sesso maschile è *rssmrx08m*.

bool salvaPerEta(const CodaRichieste* pc) salva nel file *1.txt* i dati di tutte le richieste relative a bambini di 1 anno, nel file *2.txt* tutte le richieste relative a bambini di 2 anni, e così via per tutte le età possibili. La funzione restituisce *true* se tutti i salvataggi sono andati a buon fine, *false* altrimenti. Esempio:

contenuto di 1.txt (codice, reddito, numero componenti, età)

```
brnfrn12f 23000 3 1
```

contenuto di 4.txt

```
mrtstrx09f 50000 4 4
```

```
vrdsnrx09m 30000 4 4
```

contenuto di 5.txt

```
rssmrx08m 20000 4 5
```

void stampaPerRedEq(const CodaRichieste* pc) stampa a video le richieste presenti nella coda ordinandole per reddito equivalente crescente. Il reddito equivalente è pari al reddito complessivo del nucleo familiare diviso per il numero di componenti della famiglia. Esempio:

Red. eq.	Cod. fis.
5000	rssmrx08m
7500	vrdsnrx09m
7666.67	brnfrn12f
12500	mrtstrx09f

int elimina(CodaRichieste* pc, float s) elimina dalla coda tutte le richieste aventi un reddito complessivo maggiore di *s*. Il numero di richieste rimosse è restituito al chiamante.